

PROJEKT BAZY DANYCH DOM KULTURY

**Marcin Ruta
Michał Prądziński
prowadzący: dr hab. Marcin Kowalczyk**

Dom kultury – spis treści

1. Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)

2. Definicja systemu

2.1 Perspektywy użytkowników

3. Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

3.2 Ustalenie związków między encjami i ich typów

3.3 Określenie atrybutów i ich dziedzin

3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

3.5 Klucze kandydujące i główne (decyzje projektowe)

3.6 Schemat ER na poziomie konceptualnym

3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

4. Model logiczny

4.1 Charakterystyka modelu relacyjnego

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

4.3 Proces normalizacji – analiza i przykłady

4.4 Schemat ER na poziomie modelu logicznego

4.5 Więzy integralności

4.6 Proces denormalizacji – analiza i przykłady

5. Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

5.2 Strojanie bazy danych – dobór indeksów

5.3 Skrypt SQL zakładający bazę danych

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

Bibliografia

1. Założenia i cel projektu

Realizowany projekt dotyczy domu kultury w małym mieście powiatowym. Oferuje on różne usługi związane z szeroko pojętą działalnością społeczno-kulturową m.in. Zajęcia ogólnorozwojowe, wydarzenia kulturowe.

W tym celu dom kultury posiada wiele sal w swojej lokalizacji, aby móc obsługiwać klientów/uczestników danych zajęć. Wydarzenia (baza wydarzeń) serwowane przez instytucję to między innymi zajęcia, na których uczestnik rozwija swoje pasje i uczy się nowych umiejętności np. jak śpiew, malowanie, gra na instrumencie, taniec oraz wydarzenia kulturalne skupiające większą ilość osób m.in. koncerty, wystawy, sztuki teatralne i seanse filmowe. Każda usługa jest przypisana do danego pracownika, który posiada odpowiednie kwalifikacje – nauczyciel musi mieć właściwy dyplom ukończenia kursu bądź studiów specjalistycznych, zaś organizator ma odpowiednie uprawnienia. Każdy pracownik ma swoje wynagrodzenie miesięczne.

Wydarzenia oczywiście posiadają swoich uczestników jak wcześniej zostało wspomniane. Mogą oni się zapisywać na dane zajęcia za oczywistą opłatą. Oczywiście dom kultury oferuje również wydarzenia bezpłatne. Instytucja jako organizator wydarzeń zbiera informacje o klientach w momencie zapisów m.in. adres zamieszkania, nr telefonu, imię i nazwisko.

2. Definicja systemu

2.1 Perspektywy użytkowników

- **Pracownik:** widzi swoje dane osobowe oraz historie swoich wynagrodzeń; jest w stanie zobaczyć spis zajęć, które obsługuje, oraz ma zdolność do zobaczenia uczestników danych zajęć;
- **Dyrektor:** wszystkie dane są dla niego dostępne;

Operacje na danych:

- Obszar Domu Kultury:
 - podgląd danych Domu Kultury
 - dodawanie, usuwanie i modyfikacja danych dotyczących Domu Kultury
- Obszar Pracowników:
 - dodanie lub usunięcie danego pracownika
 - modyfikacja i podgląd danych pracowników
 - przypisywanie pracowników do danego wydarzenia
 - przypisanie stanowiska do pracownika
 - podgląd listy wynagrodzeń pracowników
- Obszar Wydarzeń:
 - dodawanie, usuwanie i modyfikacja danych wydarzeń
 - ustalenie cen wydarzeń
 - przypisywanie pracowników do danego wydarzenia
 - przypisywanie uczestnika na dane wydarzenie (zapisy)

-wyliczenie zajętych miejsc na dane wydarzenie

- Obszar Uczestników:
 - podgląd danych dotyczących uczestnika
 - modyfikacja, usuwanie i dodawanie danych personalnych uczestnika
- Obszar Sal:
 - modyfikacja, usuwanie i dodawanie danych dotyczących sal
 - podgląd danych dotyczących sal

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie

Dom kultury - pojęcie zawierające szczegółowe informacje o domie kultury

Pracownik - pojęcie opisujące osobę zatrudnioną przez dom

Prowadzący - pojęcie precyzujące pracownika zatrudnioną do obsługi zajęć

Organizator - pojęcie precyzujące pracownika zatrudnioną do organizacji wydarzeń

Wydarzenia - pojęcie zawierające szczegółowe informacje o wydarzeniach

Sala - pojęcie opisujące sale znajdujące się w domu kultury

Uczestnik - pojęcie opisujące uczestnika biorącego udział w wydarzeniach

3.2. Ustalenie związków między encjami i ich typów

Dom kultury – Pracownik:

Dom kultury zatrudnia wielu pracowników lub żadnego (początek działalności).
Pracownik zaś zatrudniany jest przez jeden dom kultury.

Pracownik – Wydarzenie:

Pracownik obsługuje zero (może być już zwolniony) lub wiele wydarzeń.
Wydarzenie zaś jest obsługiwane przez jednego lub wiele pracowników.

Wydarzenie - Sala:

Wydarzenie ma miejsce w jednej sali.
Sala zaś może mieścić wiele wydarzeń lub żadnego.

Wydarzenie – Uczestnik:

Wydarzenie posiada zero lub wielu uczestników biorących w nich udział.
Uczestnik zaś może być zapisany na zero lub wiele wydarzeń.

Dom Kultury – Sala:

Dom kultury posiada jedną lub wiele sal.
Sala zaś mieści się w jednym domu kultury.

Wydarzenie – Dom Kultury:

Wydarzenie jest oferowane przez jeden dom kultury.

Dom kultury oferuje zero lub wiele wydarzeń.

Związek		Stopień związku	Typ związku	Typ uczestnictwa	
Dom kultury	Pracownik	binarny	1:n	obowiązkowy	opcjonalny
Pracownik	Wydarzenie	binarny	n:m	obowiązkowy	opcjonalny
Sala	Wydarzenie	binarny	1:n	obowiązkowy	opcjonalny
Uczestnik	Wydarzenie	binarny	n:m	opcjonalny	opcjonalny
Dom kultury	Sala	binarny	1:n	obowiązkowy	obowiązkowy
Dom kultury	Wydarzenie	binarny	1:n	obowiązkowy	opcjonalny

3.3 Określenie atrybutów i ich dziedzin**Dziedziny dla podanych argumentów:**

Dla Płci: Płec_D = {Męczyzna, Kobieta}

Dla Wykształcenia: Wykształcenie_D= {podstawowe, gimnazjalne, srednie, wyzsze, zasadnicze_zawodowe, zasadnicze_bransowe, srednie_bransowe}

Dom Kultury:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
ID_Domu	SmallInt		Tak	Tak	Numer identyfikacyjny domu kultury
Nazwa	VarChar(%p1)	30	Nie	Tak	Nazwa domu kultury
Adres	VarChar(%p1)	400	Nie	Tak	Adres domu kultury
Dyrektor	VarChar(%p1)	60	Nie	Tak	Imię i nazwisko dyrektora
Data_zalozenia	Date		Nie	Tak	Data założenia domu kultury

Pracownik:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
ID_Pracownika	SmallInt		Tak	Tak	Numer identyfikacyjny pracownika
Imie	VarChar(%p1)	20	Nie	Tak	Imię pracownika
Nazwisko	VarChar(%p1)	30	Nie	Tak	Nazwisko pracownika
Data_Urodzenia	Date		Nie	Tak	Data urodzenia pracownika
PESEL	Character(%p1)	11	Nie	Nie	Numer PESEL pracownika
Adres	VarChar(%p1)	400	Nie	Tak	Adres zamieszkania pracownika
Nr_telefonu	VarChar(%p1)	12	Nie	Nie	Numer telefonu pracownika
Email	VarChar(%p1)	50	Nie	Tak	E-mail pracownika do kontaktu
Wynagorodzenie	Money		Nie	Tak	Wynagrodzenie pracownika
Plec	VarChar(%p1), Dziedzina: Plec_D	15	Nie	Tak	Płeć pracownika
Stanowisko	VarChar(%p1)	30	Nie	Tak	Stanowisko, na którym pracownik jest zatrudniony
Data_Zatrudnienia	Date		Nie	Tak	Data zatrudnienia pracownika
Data_Zwolnienia	Date		Nie	Nie	Data zwolnienia pracownika

Prowadzący:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
Wykształcenie	VarChar(%p1), Dziedzina: Wykształcenie_D	30	Nie	Tak	Opis wykształcenia prowadzącego
Nr_Dokumentu	VarChar(%p1)	30	Nie	Tak	Numer dokumentu potwierdzającego uprawnienia
Typ_Dokumentu	VarChar(%p1)	30	Nie	Tak	Typ dokumentu uprawniającego do wykonywania zawodu
Data_Wydania_Dokumentu	Date		Nie	Tak	Data wydania dokumentu, który upoważnia do wykonywania zawodu
Specjalizacja	VarChar(%p1)	30	Nie	Tak	Specjalizacja prowadzącego

Organizator:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
Typ_Zezwolenia	VarChar(%p1)	30	Nie	Tak	Typ zezwolenia, który upoważnia do wykonywania zawodu
Nr_Zezwolenia	VarChar(%p1)	30	Nie	Tak	Numer zezwolenia, który upoważnia do wykonywania zawodu
Data_Wydania_Zezwolenia	Date		Nie	Tak	Data wydania zezwolenia
Data_Waznosci_Zezwolenia	Date		Nie	Nie	Data ważności zezwolenia

Wydarzenie:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
ID_Wydarzenia	SmallInt		Tak	Tak	Numer identyfikacyjny wydarzenia
Nazwa_Wydarzenia	VarChar(%p1)	50	Nie	Tak	Nazwa wydarzenia
Maksymalna_ilosc_uczestnikow	SmallInt		Nie	Tak	Maksymalna ilość uczestników biorąca udział w zajęciach
Cena	Money		Nie	Tak	Cena wydarzenia
Wymagana_Zgoda_Rodzica	Boolean		Nie	Tak	Czy wymagana zgoda rodzica
Data_wydarzenia	DateTime		Nie	Tak	Data wydarzenia

Uczestnik:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
ID_Uczestnika	SmallInt		Tak	Tak	Numer identyfikacyjny uczestnika
Imie	VarChar(%p1)	30	Nie	Tak	Imię uczestnika
Nazwisko	VarChar(%p1)	30	Nie	Tak	Nazwisko uczestnika
Plec	VarChar(%p1), Dziedzina: Plec_D	15	Nie	Tak	Płeć uczestnika
Data_Urodzenia	Date		Nie	Tak	Data urodzenia uczestnika
Numer_Telefonu	VarChar(%p1)	12	Nie	Nie	Numer telefonu uczestnika
Adres	VarChar(%p1)	400	Nie	Nie	Adres zamieszkania uczestnika
Email	VarChar(%p1)	50	Nie	Tak	E-mail uczestnika

Sala:

Nazwa	Typ	Parametr	Czy klucz główny	Czy obowiązkowy	Opis
ID_Sali	SmallInt		Tak	Tak	Numer identyfikacyjny sali
Numer_Sali	Character(%p1)	3	Nie	Tak	Numer sali, który znajduje się na drzwiach
Nazwa_Sali	VarChar(%p1)	40	Nie	Nie	Nazwa sali
Maksymalna_Pojemnosc	SmallInt		Nie	Tak	Maksymalna pojemność sali
Czy_Rzutnik	Boolean		Nie	Tak	Czy w sali znajduje się projektor
Czy_Internet	Boolean		Nie	Tak	Czy w sali jest internet

3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

Maksymalna pojemność sal musi być większa od zera.

Wynagrodzenie powinno być większe od zera.

Data urodzenia nie powinna być usadzona w przyszłości.

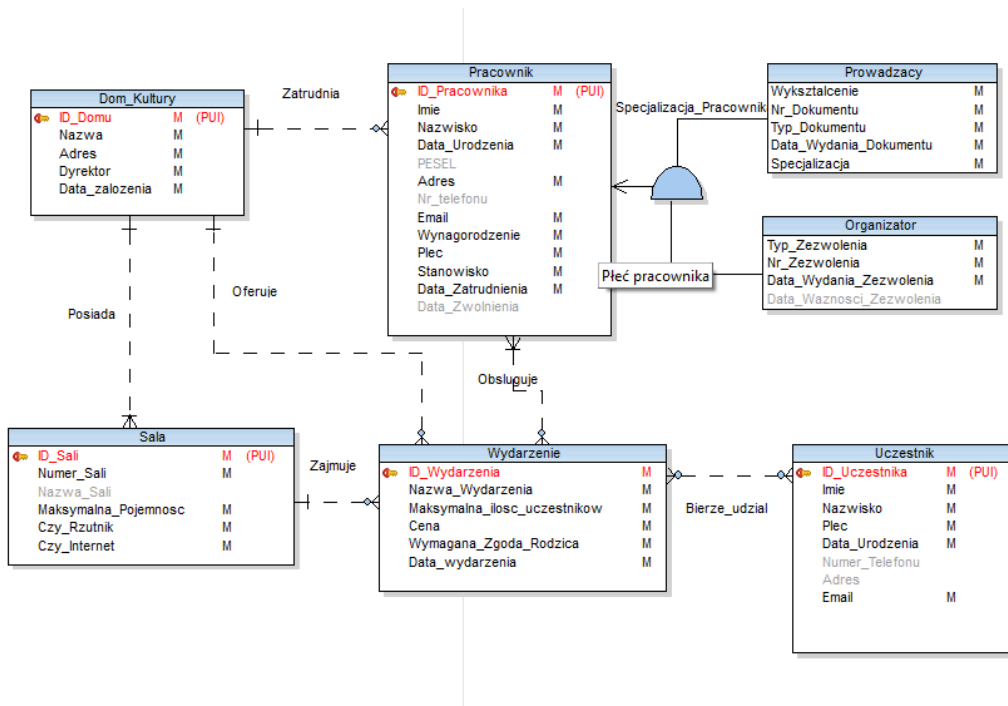
Maksymalna ilość uczestników nie może być mniejsza od zera.

Cena wydarzenia nie może być mniejsza od zera.

3.5 Klucze kandydujące i główne (decyzje projektowe)

Encja	Klucz główny	Klucz kandydujący
Dom_kultury	ID_Domu	Nazwa
Pracownik	ID_Pracownika	-
Uczestnik	ID_Uczestnika	-
Sala	ID_Sali	Nr_Sali
Wydarzenie	ID_Wydarzenia	-

3.6 Schemat ER na poziomie konceptualnym



3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

Pułapka szczelinowa:

Brak związku pomiędzy salą a Pracownikiem mógłby być pułapką szczelinową (jakbyśmy chcieli wiedzieć jaki pracownik pracuje w danej sali), lecz z naszego punktu widzenia nie jest nią, ponieważ pracownik jest dopiero przypisany do sali jak obsługuje dane wydarzenie. Także nie istnieje sytuacja, kiedy pracownik bez wydarzenia jest przypisany do sali.

Kolejna możliwa pułapka szczelinowa zaistniałaby, kiedy chcielibyśmy zdobyć informacje o uczestniku, który już nie jest zapisany na żadne wydarzenie. W takim wypadku nie ma jak się dostać do tych informacji przez encję wydarzenie, więc sugerowany jest związek między domem kultury a uczestnikiem. Jednakże nie jest nam to potrzebne, jeżeli uwzględnimy modelowanie czasu w związku wydarzenie – uczestnik.

Pułapka wachlarzowa:

Nie zauważyliśmy w naszym modelu żadnej pułapki. Wystąpiłaby, gdyby nie było połączenia pomiędzy Pracownikiem a Wydarzeniem. W takim przypadku byłyby związki pomiędzy Domem Kultury i Pracownikiem oraz Domem Kultury i Wydarzeniem. Z takich związków nie moglibyśmy się dowiedzieć, który pracownik obsługuje dane wydarzenie.

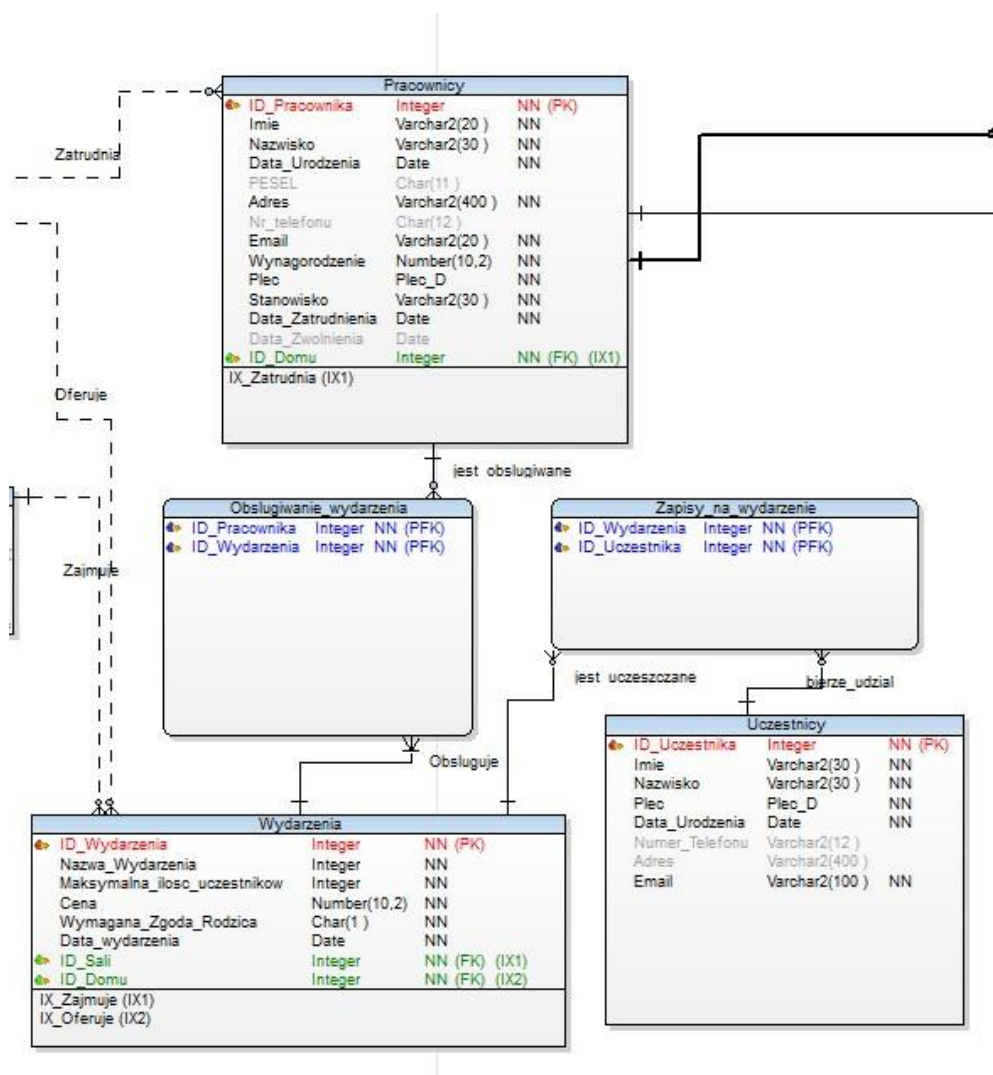
4. Model logiczny

4.1 Charakterystyka modelu relacyjnego

W celu otrzymania modelu relacyjnego, skorzystaliśmy z generatora zawartego w narzędziu TOAD Data Modeler. W ramach tego, generator usunął związki wielu do wielu i zastąpił je związkami jeden do wielu do tablicy bridge'ującej oraz wstawił odpowiednie klucze główne do encji, w których nie było owego (w naszym modelu nie było go w encjach precyzujących pracownika – organizator i prowadzący). Dodatkowo, aplikacja nie wspomaga języka polskiego, dlatego należało zmienić nazwy relacji na liczbę mnogą. Kolejnym krokiem było zastąpienie pól segmentowych odpowiednimi relacjami oraz związkami między starymi relacjami a nowo utworzonymi.

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Aby uniknąć niekompatybilności z modelem relacyjnym, należy usunąć związki wielu do wielu i zastąpić je związkami jeden do wielu do tablicy łączącej. W naszym projekcie były takie związki dwa: Pracownik – Wydarzenie oraz Uczestnik – Wydarzenie.



4.3 Proces normalizacji – analiza i przykłady

4.3.1 Pierwsza postać normalna

Najważniejszym elementem procesu normalizacji było przejście do pierwszej postaci normalnej. Warunkiem jej jest brak istnienia pól segmentowych (wartości atomowe) oraz brak powtarzających się grup. W naszym modelu relacyjnym występowało kilka takich przypadków. Pierwszym z nich jest pole Adres - wstępuje on w wielu tabelach i jest to pole segmentowe. Stworzyliśmy z tego względu nową relację Adresy. Dodatkowo stworzyliśmy osobną jeszcze relację Poczty, ponieważ kod pocztowy i miasto to powtarzające się grupy.

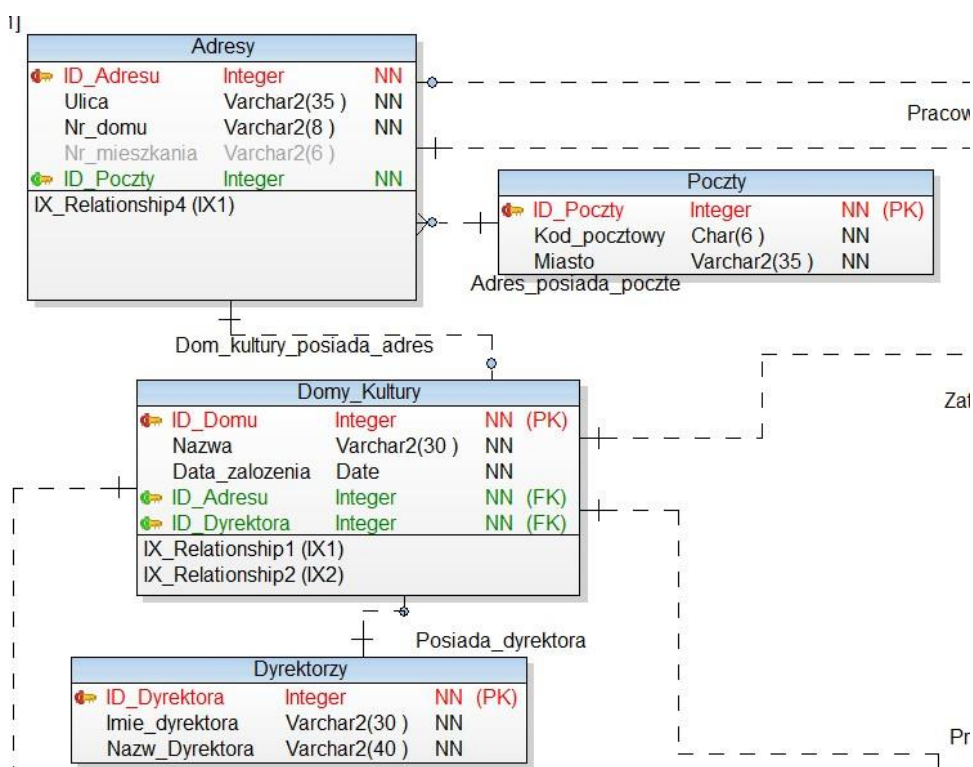
Kolejnym przykładem normalizacji w naszym projekcie było stworzenie nowej encji Dyrektorzy, ze względu na wcześniej występujące w encji pole segmentowe. Należało więc rozdzielić imię i nazwisko jako osobne atrybuty.

Proces normalizacji pokazany jest na zdjęciu poniżej.

Przed normalizacją:

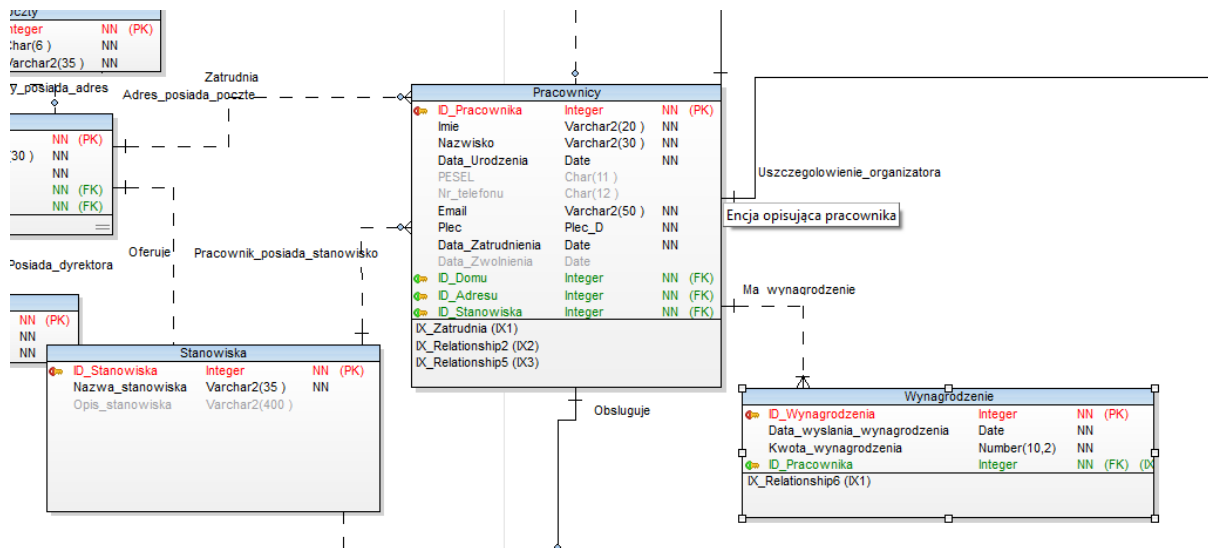


Po normalizacji:



Następnym przykładem normalizacji jest stworzenie nowych relacji Stanowiska oraz Wynagrodzenie. W pierwszym przypadku należało stworzyć encję Stanowiska, ponieważ są to powtarzające się grupy - uniknęliśmy w ten sposób redundancji danych. W wypadku tabeli Wynagrodzenia, zdecydowaliśmy się również na taki ruch, ponieważ było to pole wielowartościowe w relacji pracownicy.

Zdjęcie ukazujące efekt po normalizacji:

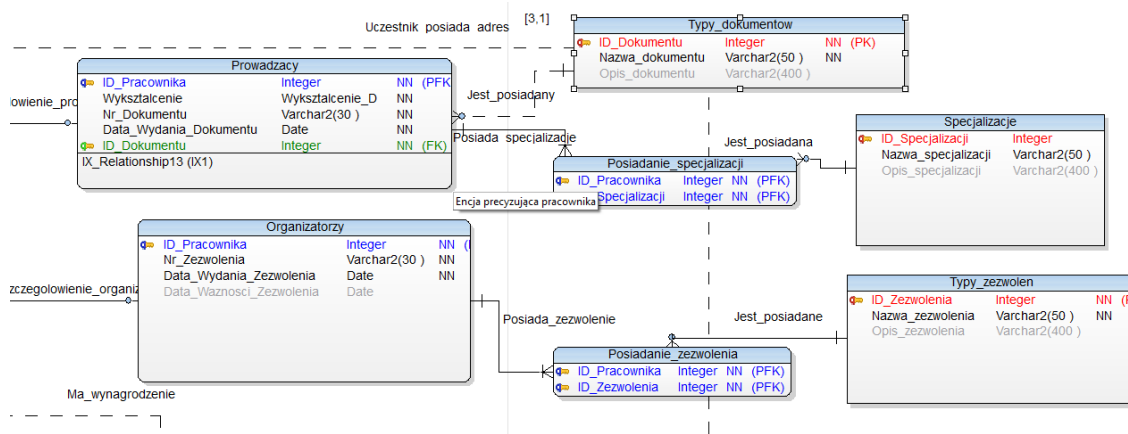


Kolejną rzeczą, którą wzięliśmy pod uwagę była relacja Prowadzący. Atrybut typ_dokumentu był powtarzającą się grupą, dlatego zdecydowaliśmy się na stworzenie nowej tabeli Typy_dokumentow.

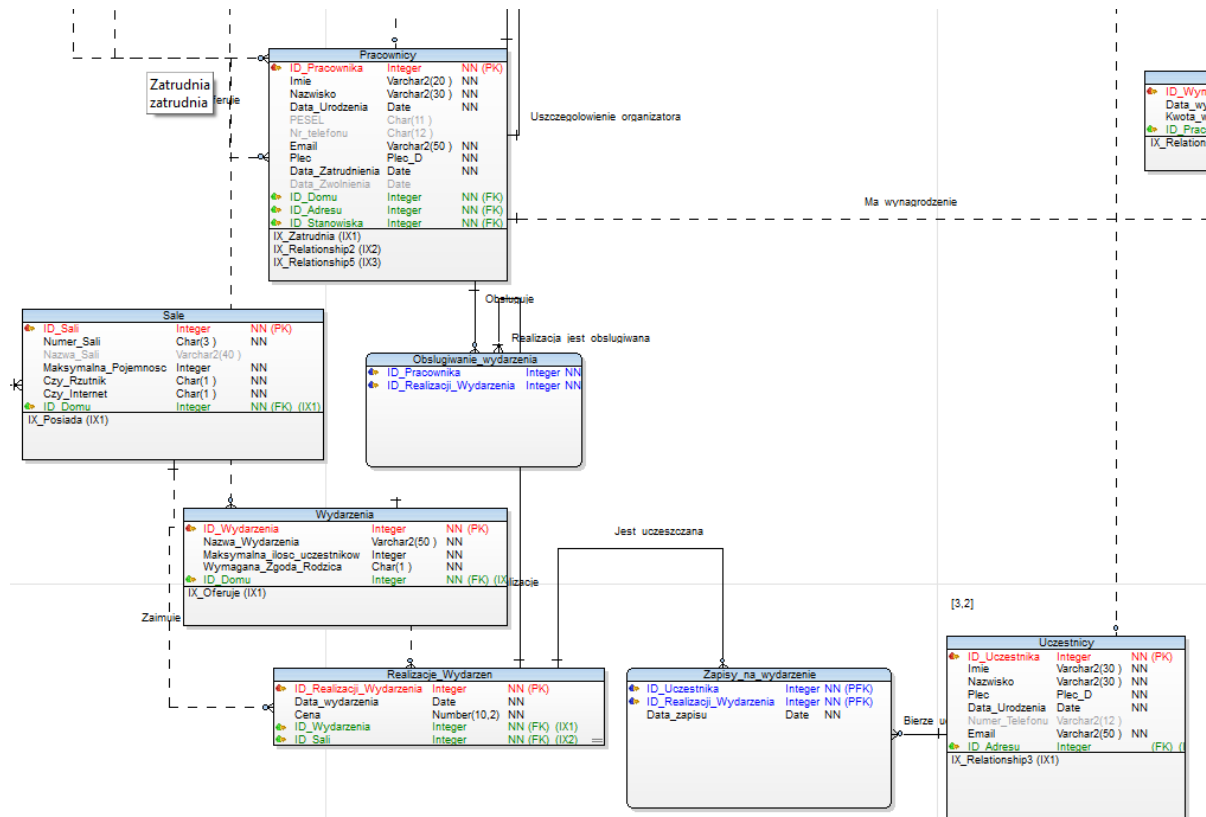
Atrybut specjalizacja również w naszym projekcie jest powtarzającą się grupą, jednakże wystąpiłby pomiędzy Prowadzącymi a Specjalizacjami związek wielu do wielu (prowadzący może posiadać wiele specjalizacji, a specjalizacje może posiadać wiele prowadzących), dlatego skorzystaliśmy z tablicy łączącej Posiadanie_specjalizacji (oraz związkami jeden do wielu).

Bardzo podobnym przypadkiem jest atrybut typ_zwolenia. Również wystąpiłby tutaj związek wielu do wielu pomiędzy nowo powstałą relacją Typy_zezwolen a Organizatorami, który zastąpiliśmy tablicą łączącą Posiadanie_zezwolenia.

Poniżej znajduje się znormalizowana postać modelu:



Ostatnim przykładem normalizacji było stworzenie relacji Realizacja_Wydarzen, ze względu na wielowartościowość pola Data i zależnego od niego pola cena. Następnie zdecydowaliśmy się, że związki pomiędzy Uczestnikami, Pracownikami (przez tablice łączące), Salami a Wydarzeniami zastąpimy związkami między powyżej wymienionymi relacjami a Realizacjami_Wydarzen, ponieważ chcemy uzyskać powiązania pomiędzy daną realizacją wydarzenia, a nie ogólnym wydarzeniem.



4.3.2 Druga postać normalna

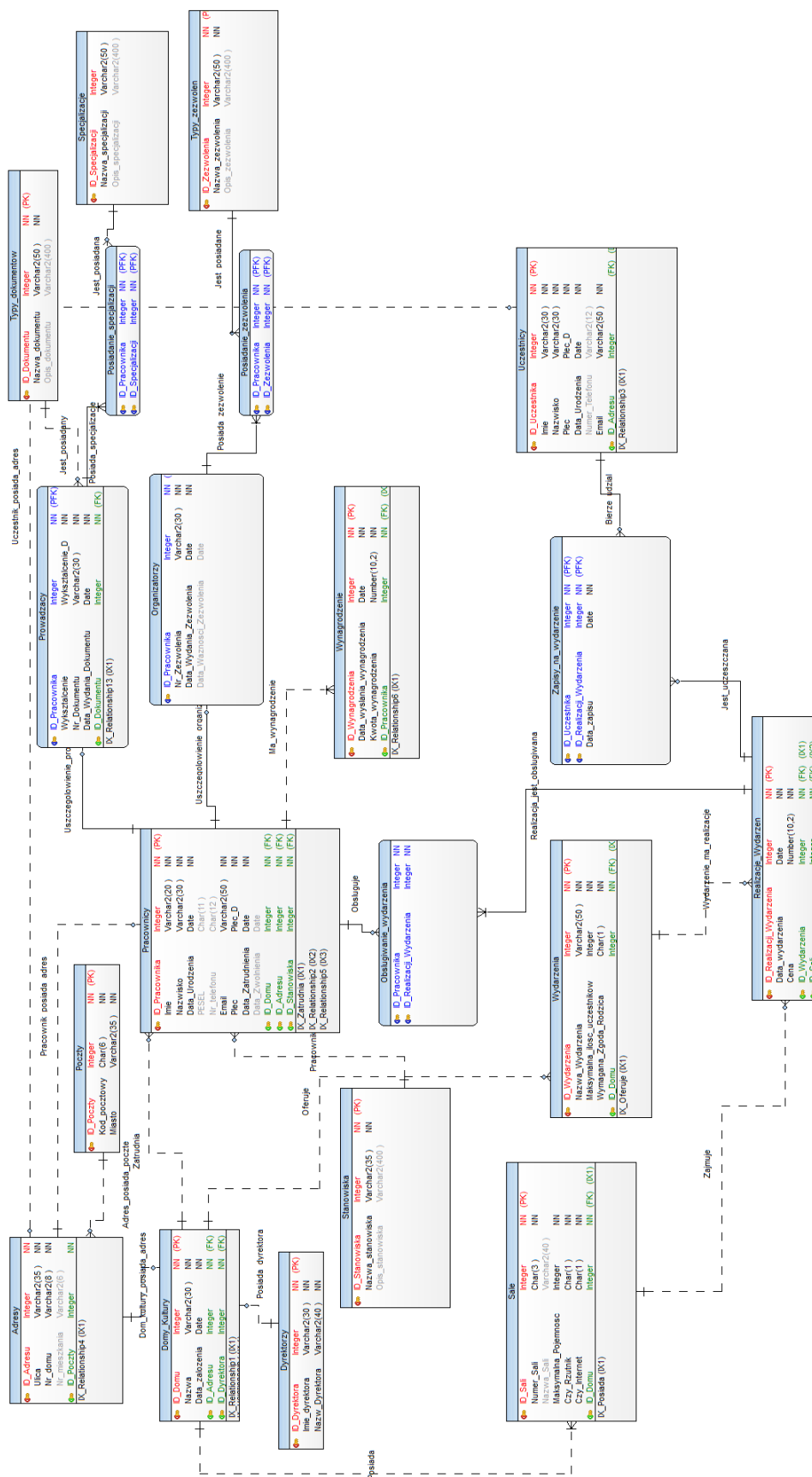
W naszym modelu wszystkie klucze potencjalne są kluczami prostymi. W takim wypadku, gdy model zaczął spełniać wymogi pierwszej postaci normalnej, zaczął również spełniać kryteria drugiej postaci normalnej.

4.3.3 Trzecia postać normalna

W naszym projekcie atrybuty danych relacji, które nie wchodzą w skład żadnego klucza potencjalnego, nie są przechodnio funkcyjnie zależne od żadnego klucza potencjalnego danej relacji.

Na etapie projektowania naszej bazy danych pamiętaliśmy o założeniach postaci normalnych, dlatego gdy już wcześniej rozbijaliśmy pola segmentowe i wielowartościowe na odpowiednie relacje oraz pola z powtarzającymi się wartościami na relacje słownikowe, pozbyliśmy się w taki sposób zależności przechodnio funkcyjnych.

4.4 Schemat ER na poziomie modelu logicznego



4.5 Więzy integralności

W naszym projekcie nie występują powtarzające się klucze, więc każdy z nich jest unikalny. Dodatkowo nie ma pól segmentowych, a większość atrybutów w relacjach będzie musiało posiadać jakąś wartość (NOT NULL).

4.6 Proces denormalizacji – analiza i przykłady

Podczas procesu denormalizacji zastanawialiśmy się nad kilkoma rozwiązaniami. Pierwszym z nich było usunięcie relacji Dyrektorzy ze względu na to, że nie będziemy w bazie danych używać zapytania “jak na imię ma dyrektor”, lecz interesuje nas całość - imię oraz nazwisko. Dodatkowo relacja dyrektor nie wnosi żadnych dodatkowych informacji, dlatego zdecydowaliśmy się na powrót atrybutu Dyrektor do relacji Domy_Kultury.

W przypadku tabel Poczty oraz Adres, zastanawialiśmy się nad ponownym połączeniem obydwu tabel, ponieważ częstym zapytaniem do bazy będzie pytanie o adres cały - baza będzie musiała skorzystać z operacji JOIN, która może nie być zbyt efektywna. Jednakże zależy nam nad zapobieganiem redundancji danych (dla lokalnego domu kultury poczty bardzo często będą się powtarzać), dlatego zdecydowaliśmy się nad pozostawieniem osobnej tabeli Poczty.

Z podobnego punktu widzenia zostawiliśmy tabele Typy_dokumentow, Specjalizacje oraz Typy_zezwolen – bardzo często dane będą się powtarzać a ich całkowita ilość nie będzie przytłaczająco duża.

W załączniku 6.1 znajduje się zdjęcie modelu po denormalizacji.

5. Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Zasoby	Czy wykonywalna?
Podgląd danych Domu Kultury	Domy_kultury	TAK
Dodawanie, usuwanie i modyfikacja danych dotyczących Domu Kultury	Domy_kultury	TAK
Dodatnie lub usunięcie danego pracownika	Pracownicy	TAK
Modyfikacja i podgląd danych pracowników	Pracownicy	TAK
Przypisywanie pracowników do danego wydarzenia	Pracownicy, Realizacje_Wydarzen	TAK
Przypisanie stanowiska do pracownika	Stanowiska, Pracownicy	TAK
Podgląd listy wynagrodzeń pracowników	Wynagrodzenia, Pracownicy	TAK
Dodawanie, usuwanie i modyfikacja danych wydarzeń	Wydarzenia, Realizacja_Wydarzen	TAK

Ustalenie cen wydarzeń	Realizacje_wydarzen	TAK
Przypisywanie pracowników do danego wydarzenia	Obslugiwanie_wydarzenia, Pracownicy, Realizacja_wydarzen	TAK
Przypisywanie uczestnika na dane wydarzenie (zapisy)	Uczestnicy, Realizacja_Wydarzen	TAK
Wyliczenie zajętych miejsc na dane wydarzenie	Zapisy_na_wydarzenie, Uczestnicy, Realizacje_wydarzen	TAK
Podgląd danych dotyczących uczestnika	Uczestnicy	TAK
Modyfikacja, usuwanie i dodawanie danych personalnych uczestnika	Uczestnicy	TAK
Modyfikacja, usuwanie i dodawanie danych dotyczących sal	Sale	TAK
Podgląd danych dotyczących sal	Sale	TAK

5.2 Strojenie bazy danych – dobór indeksów

Dom_Kultury:

```
CREATE INDEX IX_Adres_posiada ON Domy_Kultury (ID_Adresu)
```

Adresy:

```
CREATE INDEX IX_Posiada_poczte ON Adresy (ID_Poczty)
```

Wynagrodzenia:

```
CREATE INDEX IX_Pracownik_otrzymuje ON Wynagrodzenia (ID_Pracownika)
```

Pracownicy:

```
CREATE INDEX IX_Zatrudnia ON Pracownicy (ID_Domu)
```

```
CREATE INDEX IX_Adres_jest_posiadany ON Pracownicy (ID_Adresu)
```

```
CREATE INDEX IX_Stanowisko_posiada ON Pracownicy (ID_Stanowiska)
```

Sale:

```
CREATE INDEX IX_Dom_posiada ON Sale (ID_Domu)
```

Wydarzenia:

```
CREATE INDEX IX_Oferuje ON Wydarzenia (ID_Domu)
```

Realizacje_Wydarzen:

```
CREATE INDEX IX_Wydarzenie_ma ON Realizacje_Wydarzen (ID_Wydarzenia)
```

```
CREATE INDEX IX_Sala_posiada ON Realizacje_Wydarzen (ID_Sali)
```

Uczestnicy:

```
CREATE INDEX IX_Adres_ma ON Uczestnicy (ID_Adresu)
```

Prowadzacy:

```
CREATE INDEX IX_Jest_posiadany ON Prowadzacy (ID_Dokumentu)
```

5.3 Skrypt SQL zakładający bazę danych

5.3.1. Skrypt wygenerowany przez TOAD Data Modeler

```
/*
Created: 12.11.2020
Modified: 17.11.2020
Project: Dom Kultury
Model: Dom_kultury
Company: Politechnika Warszawska
Author: Michał Prądziński, Marcin Ruta
Version: 1.4
Database: Oracle 12c
*/

-- Create sequences section -----

CREATE SEQUENCE DomSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE AdresSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE PocztySeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE PracownikSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE DokumentSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE ZezwolenieSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE SpecjalizacjeSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE UczestnikSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE StanowiskoSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE SalaSeq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/
```

```
CREATE SEQUENCE WydarzenieSeq1
```

```

    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE RealizacjaSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE WynagrodzenieSeq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

-- Create tables section -----

-- Table Domy_Kultury

CREATE TABLE Domy_Kultury(
    ID_Domu Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Dyrektor Varchar2(60 ) NOT NULL,
    ID_Adresu Integer NOT NULL
)
/

-- Create indexes for table Domy_Kultury

CREATE INDEX IX_Adres_posiada ON Domy_Kultury (ID_Adresu)
/

-- Add keys for table Domy_Kultury

ALTER TABLE Domy_Kultury ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY (ID_Domu)
/

-- Table and Columns comments section

COMMENT ON COLUMN Domy_Kultury.Dyrektor IS 'Imie i nazwisko dyrektora'
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    ID_Pracownika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,

```

```

    Data_Urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Nr_telefonu Varchar2(12 ),
    Email Varchar2(50 ) NOT NULL,
    Plec Varchar2(15 ) NOT NULL
        CHECK (Plec IN ('Meczczyzna', 'Kobieta')),
    Data_Zatrudnienia Date NOT NULL,
    Data_Zwolnienia Date,
    ID_Domu Integer NOT NULL,
    ID_Adresu Integer NOT NULL,
    ID_Stnowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_Zatrudnia ON Pracownicy (ID_Domu)
/

CREATE INDEX IX_Adres_jest_posiadany ON Pracownicy (ID_Adresu)
/

CREATE INDEX IX_Stnowisko_posiada ON Pracownicy (ID_Stnowiska)
/

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY
(ID_Pracownika)
/

-- Table Prowadzacy

CREATE TABLE Prowadzacy(
    ID_Pracownika Integer NOT NULL,
    Wyksztalcenie Varchar2(35 ) NOT NULL
        CHECK (Wyksztalcenie IN
('podstawowe', 'gimnazjalne', 'srednie', 'wyzsze', 'zasadnicze_zawodowe', 'zasadnicze_br
anzowe', 'srednie_branzowe')),
    Nr_Dokumentu Varchar2(30 ) NOT NULL,
    Data_Wydania_Dokumentu Date NOT NULL,
    ID_Dokumentu Integer NOT NULL
)
/

-- Create indexes for table Prowadzacy

CREATE INDEX IX_Jest_posiadany ON Prowadzacy (ID_Dokumentu)
/

-- Add keys for table Prowadzacy

ALTER TABLE Prowadzacy ADD CONSTRAINT Unique_Identifier3 PRIMARY KEY
(ID_Pracownika)
/

-- Table Organizatorzy

```

```

CREATE TABLE Organizatorzy(
    ID_Pracownika Integer NOT NULL,
    Nr_Zezwolenia Varchar2(30 ) NOT NULL,
    Data_Wydania_Zezwolenia Date NOT NULL,
    Data_Waznosci_Zezwolenia Date
)
/

-- Add keys for table Organizatorzy

ALTER TABLE Organizatorzy ADD CONSTRAINT Unique_Identifier4 PRIMARY KEY
(ID_Pracownika)
/

-- Table Wydarzenia

CREATE TABLE Wydarzenia(
    ID_Wydarzenia Integer NOT NULL,
    Nazwa_Wydarzenia Varchar2(50 ) NOT NULL,
    Maksymalna_ilosc_uczestnikow Integer NOT NULL,
    Wymagana_Zgoda_Rodzica Char(1 ) NOT NULL,
    ID_Domu Integer NOT NULL
)
/

-- Create indexes for table Wydarzenia

CREATE INDEX IX_Oferuje ON Wydarzenia (ID_Domu)
/

-- Add keys for table Wydarzenia

ALTER TABLE Wydarzenia ADD CONSTRAINT Unique_Identifier5 PRIMARY KEY
(ID_Wydarzenia)
/

-- Table Uczestnicy

CREATE TABLE Uczestnicy(
    ID_Uczestnika Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Plec Varchar2(15 ) NOT NULL
        CHECK (Plec IN ('Meczczyzna', 'Kobieta')),
    Data_Urodzenia Date NOT NULL,
    Numer_Telefonu Varchar2(12 ),
    Email Varchar2(50 ) NOT NULL,
    ID_Adresu Integer
)
/

-- Create indexes for table Uczestnicy

CREATE INDEX IX_Adres_ma ON Uczestnicy (ID_Adresu)
/

```

```

-- Add keys for table Uczestnicy

ALTER TABLE Uczestnicy ADD CONSTRAINT Unique_Identifier6 PRIMARY KEY
(ID_Uczestnika)
/

-- Table Sale

CREATE TABLE Sale(
    ID_Sali Integer NOT NULL,
    Numer_Sali Char(3 ) NOT NULL,
    Nazwa_Sali Varchar2(40 ),
    Maksymalna_Pojemnosc Integer NOT NULL,
    Czy_Rzutnik Char(1 ) NOT NULL,
    Czy_Internet Char(1 ) NOT NULL,
    ID_Domu Integer NOT NULL
)
/

-- Create indexes for table Sale

CREATE INDEX IX_Dom_posiada ON Sale (ID_Domu)
/

-- Add keys for table Sale

ALTER TABLE Sale ADD CONSTRAINT Unique_Identifier7 PRIMARY KEY (ID_Sali)
/

-- Table Obslugiwanie_wydarzenia

CREATE TABLE Obslugiwanie_wydarzenia(
    ID_Pracownika Integer NOT NULL,
    ID_Realizacji_Wydarzenia Integer NOT NULL
)
/

-- Table Zapisy_na_wydarzenie

CREATE TABLE Zapisy_na_wydarzenie(
    ID_Uczestnika Integer NOT NULL,
    ID_Realizacji_Wydarzenia Integer NOT NULL,
    Data_zapisu Date NOT NULL
)
/

-- Table and Columns comments section

COMMENT ON COLUMN Zapisy_na_wydarzenie.Data_zapisu IS 'Data zapisu na dane zajęcia'
/

-- Table Adresy

CREATE TABLE Adresy(
    ID_Adresu Integer NOT NULL,
    Ulica Varchar2(35 ) NOT NULL,
    Nr_domu Varchar2(8 ) NOT NULL,

```

```

        Nr_mieszkania Varchar2(6),
        ID_Poczty Integer NOT NULL
    )
/

-- Create indexes for table Adresy

CREATE INDEX IX_Posiada_poczte ON Adresy (ID_Poczty)
/

-- Add keys for table Adresy

ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (ID_Adresu)
/

-- Table and Columns comments section

COMMENT ON COLUMN Adresy.ID_Adresu IS 'Numer identyfikacyjny poczty'
/
COMMENT ON COLUMN Adresy.Ulica IS 'Nazwa ulicy'
/
COMMENT ON COLUMN Adresy.Nr_domu IS 'Numer domu'
/
COMMENT ON COLUMN Adresy.Nr_mieszkania IS 'Numer mieszkania'
/

-- Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
    ID_Wynagrodzenia Integer NOT NULL,
    Data_wyslania_wynagrodzenia Date NOT NULL,
    Kwota_wynagrodzenia Number(10,2) NOT NULL,
    ID_Pracownika Integer NOT NULL
)
/

-- Create indexes for table Wynagrodzenia

CREATE INDEX IX_Pracownik_otrzymuje ON Wynagrodzenia (ID_Pracownika)
/

-- Add keys for table Wynagrodzenia

ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia PRIMARY KEY
(ID_Wynagrodzenia)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wynagrodzenia.ID_Wynagrodzenia IS
'Numer identyfikacyjny wynagrodzenia'
/
COMMENT ON COLUMN Wynagrodzenia.Kwota_wynagrodzenia IS 'Kwota wynagrodzenia'
/

-- Table Poczty

```



```

CREATE TABLE Poczty(
    ID_Poczty Integer NOT NULL,
    Kod_pocztowy Char(6 ) NOT NULL,
    Miasto Varchar2(35 ) NOT NULL
)
/

-- Add keys for table Poczty

ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (ID_Poczty)
/

-- Table and Columns comments section

COMMENT ON COLUMN Poczty.ID_Poczty IS 'Numer identyfikacyjny poczty'
/
COMMENT ON COLUMN Poczty.Kod_pocztowy IS 'Kod pocztowy'
/
COMMENT ON COLUMN Poczty.Miasto IS 'Miasto poczty'
/

-- Table Stanowiska

CREATE TABLE Stanowiska(
    ID_Stnowiska Integer NOT NULL,
    Nazwa_stnowiska Varchar2(35 ) NOT NULL,
    Opis_stnowiska Varchar2(400 )
)
/

-- Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stnowiska PRIMARY KEY (ID_Stnowiska)
/

-- Table and Columns comments section

COMMENT ON TABLE Stanowiska IS 'Encja opisujaca stanowska '
/
COMMENT ON COLUMN Stanowiska.ID_Stnowiska IS 'Numer identyfikacyjny stanowska'
/
COMMENT ON COLUMN Stanowiska.Nazwa_stnowiska IS 'Nazwa stanowska'
/
COMMENT ON COLUMN Stanowiska.Opis_stnowiska IS 'Opis stanowska'
/

-- Table Typy_zezwolen

CREATE TABLE Typy_zezwolen(
    ID_Zezwolenia Integer NOT NULL,
    Nazwa_zezwolenia Varchar2(50 ) NOT NULL,
    Opis_zezwolenia Varchar2(400 )
)
/

-- Add keys for table Typy_zezwolen

```

```

ALTER TABLE Typy_zezwolen ADD CONSTRAINT PK_Typy_zezwolen PRIMARY KEY
(ID_Zezwolenia)
/

-- Table and Columns comments section

COMMENT ON COLUMN Typy_zezwolen.ID_Zezwolenia IS 'Numer identyfikacyjny
zezwolenia'
/
COMMENT ON COLUMN Typy_zezwolen.Nazwa_zezwolenia IS 'Nazwa zezwolenia, który
upoważnia do wykonywania profesji'
/

-- Table Posiadanie_zezwolenia

CREATE TABLE Posiadanie_zezwolenia(
    ID_Pracownika Integer NOT NULL,
    ID_Zezwolenia Integer NOT NULL
)
/

-- Add keys for table Posiadanie_zezwolenia

ALTER TABLE Posiadanie_zezwolenia ADD CONSTRAINT PK_Posiadanie_zezwolenia PRIMARY
KEY (ID_Pracownika, ID_Zezwolenia)
/

-- Table Specjalizacje

CREATE TABLE Specjalizacje(
    ID_Specjalizacji Integer NOT NULL,
    Nazwa_specjalizacji Varchar2(50 ) NOT NULL,
    Opis_specjalizacji Varchar2(400 )
)
/

-- Add keys for table Specjalizacje

ALTER TABLE Specjalizacje ADD CONSTRAINT PK_Specjalizacje PRIMARY KEY
(ID_Specjalizacji)
/

-- Table and Columns comments section

COMMENT ON COLUMN Specjalizacje.ID_Specjalizacji IS 'Numer identyfikacyjny
specjalizacji'
/
COMMENT ON COLUMN Specjalizacje.Nazwa_specjalizacji IS 'Nazwa specjalizacji, jaką
się zajmuje'
/
COMMENT ON COLUMN Specjalizacje.Opis_specjalizacji IS 'Opis specjalizacji'
/

-- Table Posiadanie_specjalizacji

CREATE TABLE Posiadanie_specjalizacji(
    ID_Pracownika Integer NOT NULL,

```

```

        ID_Specjalizacji Integer NOT NULL
    )
/

-- Add keys for table Posiadanie_specjalizacji

ALTER TABLE Posiadanie_specjalizacji ADD CONSTRAINT PK_Posiadanie_specjalizacji
PRIMARY KEY (ID_Pracownika, ID_Specjalizacji)
/

-- Table Typy_dokumentow

CREATE TABLE Typy_dokumentow(
    ID_Dokumentu Integer NOT NULL,
    Nazwa_dokumentu Varchar2(50 ) NOT NULL,
    Opis_dokumentu Varchar2(400 )
)
/

-- Add keys for table Typy_dokumentow

ALTER TABLE Typy_dokumentow ADD CONSTRAINT PK_Typy_dokumentow PRIMARY KEY
(ID_Dokumentu)
/

-- Table and Columns comments section

COMMENT ON COLUMN Typy_dokumentow.ID_Dokumentu IS 'Numer indentyfikacyjny
dokumentu'
/
COMMENT ON COLUMN Typy_dokumentow.Nazwa_dokumentu IS 'Nazwa dokumentu, który
upoważnia do wykonywania danego zawodu'
/
COMMENT ON COLUMN Typy_dokumentow.Opis_dokumentu IS 'Opis dokumentu'
/

-- Table Realizacje_Wydarzen

CREATE TABLE Realizacje_Wydarzen(
    ID_Realizacji_Wydarzenia Integer NOT NULL,
    Data_wydarzenia Date NOT NULL,
    Cena Number(10,2) NOT NULL,
    ID_Wydarzenia Integer NOT NULL,
    ID_Sali Integer NOT NULL
)
/

-- Create indexes for table Realizacje_Wydarzen

CREATE INDEX IX_Wydarzenie_ma ON Realizacje_Wydarzen (ID_Wydarzenia)
/

CREATE INDEX IX_Sala_posiada ON Realizacje_Wydarzen (ID_Sali)
/

-- Add keys for table Realizacje_Wydarzen

```

```

ALTER TABLE Realizacje_Wydarzen ADD CONSTRAINT PK_Realizacje_Wydarzen PRIMARY KEY
(ID_Realizacji_Wydarzenia)
/

-- Table and Columns comments section

COMMENT ON COLUMN Realizacje_Wydarzen.ID_Realizacji_Wydarzenia IS 'Numer
identyfikacyjny realizacji danego wydarzenia'
/
COMMENT ON COLUMN Realizacje_Wydarzen.Data_wydarzenia IS 'Data wydarzenia'
/
COMMENT ON COLUMN Realizacje_Wydarzen.Cena IS 'Cena wydarzenia'
/

-- Trigger for sequence DomSeq1 for column ID_Domu in table Domy_Kultury -----
CREATE OR REPLACE TRIGGER ts_Domy_Kultury_DomSeq1 BEFORE INSERT
ON Domy_Kultury FOR EACH ROW
BEGIN
    :new.ID_Domu := DomSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Domy_Kultury_DomSeq1 AFTER UPDATE OF ID_Domu
ON Domy_Kultury FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Domu in table
Domy_Kultury as it uses sequence. ');
END;
/

-- Trigger for sequence PracownikSeq1 for column ID_Pracownika in table Pracownicy
-----
CREATE OR REPLACE TRIGGER ts_Pracownicy_PracownikSeq1 BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN
    :new.ID_Pracownika := PracownikSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_PracownikSeq1 AFTER UPDATE OF
ID_Pracownika
ON Pracownicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Pracownika in table
Pracownicy as it uses sequence. ');
END;
/

-- Trigger for sequence WydarzenieSeq1 for column ID_Wydarzenia in table Wydarzenia
-----
CREATE OR REPLACE TRIGGER ts_Wydarzenia_WydarzenieSeq1 BEFORE INSERT
ON Wydarzenia FOR EACH ROW
BEGIN
    :new.ID_Wydarzenia := WydarzenieSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wydarzenia_WydarzenieSeq1 AFTER UPDATE OF
ID_Wydarzenia
ON Wydarzenia FOR EACH ROW

```

```

BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Wydarzenia in table
Wydarzenia as it uses sequence.');
```

END;

/

-- Trigger for sequence UczestnikSeq1 for column ID_Uczestnika in table Uczestnicy -----

```

CREATE OR REPLACE TRIGGER ts_Uczestnicy_UczestnikSeq1 BEFORE INSERT
ON Uczestnicy FOR EACH ROW
BEGIN
    :new.ID_Uczestnika := UczestnikSeq1.nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER tsu_Uczestnicy_UczestnikSeq1 AFTER UPDATE OF
ID_Uczestnika
ON Uczestnicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Uczestnika in table
Uczestnicy as it uses sequence.');
```

END;

/

-- Trigger for sequence SalaSeq1 for column ID_Sali in table Sale -----

```

CREATE OR REPLACE TRIGGER ts_Sale_SalaSeq1 BEFORE INSERT
ON Sale FOR EACH ROW
BEGIN
    :new.ID_Sali := SalaSeq1.nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER tsu_Sale_SalaSeq1 AFTER UPDATE OF ID_Sali
ON Sale FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Sali in table Sale as it
uses sequence.');
```

END;

/

-- Trigger for sequence AdresSeq1 for column ID_Adresu in table Adresy -----

```

CREATE OR REPLACE TRIGGER ts_Adresy_AdresSeq1 BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
    :new.ID_Adresu := AdresSeq1.nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER tsu_Adresy_AdresSeq1 AFTER UPDATE OF ID_Adresu
ON Adresy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Adresu in table Adresy as
it uses sequence.');
```

END;

/

-- Trigger for sequence WynagrodzenieSeq1 for column ID_Wynagrodzenia in table Wynagrodzenia -----

```

CREATE OR REPLACE TRIGGER ts_Wynagrodzenia_Wynagrodzen_0 BEFORE INSERT
ON Wynagrodzenia FOR EACH ROW
```

```

BEGIN
    :new.ID_Wynagrodzenia := WynagrodzenieSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wynagrodzenia_Wynagrodze_0 AFTER UPDATE OF
ID_Wynagrodzenia
ON Wynagrodzenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Wynagrodzenia in table
Wynagrodzenia as it uses sequence.');
```

```

END;
/

-- Trigger for sequence PocztySeq1 for column ID_Poczty in table Poczty -----
CREATE OR REPLACE TRIGGER ts_Poczty_PocztySeq1 BEFORE INSERT
ON Poczty FOR EACH ROW
BEGIN
    :new.ID_Poczty := PocztySeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Poczty_PocztySeq1 AFTER UPDATE OF ID_Poczty
ON Poczty FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Poczty in table Poczty as
it uses sequence.');
```

```

END;
/

-- Trigger for sequence StanowiskoSeq1 for column ID_Stanowiska in table Stanowiska
-----
CREATE OR REPLACE TRIGGER ts_Stanowiska_StanowiskoSeq1 BEFORE INSERT
ON Stanowiska FOR EACH ROW
BEGIN
    :new.ID_Stanowiska := StanowiskoSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Stanowiska_StanowiskoSeq1 AFTER UPDATE OF
ID_Stanowiska
ON Stanowiska FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Stanowiska in table
Stanowiska as it uses sequence.');
```

```

END;
/

-- Trigger for sequence ZezwolenieSeq1 for column ID_Zezwolenia in table
Typy_zezwolen -----
CREATE OR REPLACE TRIGGER ts_Typy_zezwolen_ZezwolenieS_0 BEFORE INSERT
ON Typy_zezwolen FOR EACH ROW
BEGIN
    :new.ID_Zezwolenia := ZezwolenieSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Typy_zezwolen_Zezwolenie_0 AFTER UPDATE OF
ID_Zezwolenia
ON Typy_zezwolen FOR EACH ROW
BEGIN
```

```

        RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Zezwolenia in table
Typy_zezwolen as it uses sequence.');
```

END;

/

```

-- Trigger for sequence SpecjalizacjeSeq1 for column ID_Specjalizacji in table
Specjalizacje -----
CREATE OR REPLACE TRIGGER ts_Specjalizacje_Specjalizac_0 BEFORE INSERT
ON Specjalizacje FOR EACH ROW
BEGIN
    :new.ID_Specjalizacji := SpecjalizacjeSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Specjalizacje_Specjaliza_0 AFTER UPDATE OF
ID_Specjalizacji
ON Specjalizacje FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Specjalizacji in table
Specjalizacje as it uses sequence.');
```

END;

/

```

-- Trigger for sequence DokumentSeq1 for column ID_Dokumentu in table
Typy_dokumentow -----
CREATE OR REPLACE TRIGGER ts_Typy_dokumentow_DokumentS_0 BEFORE INSERT
ON Typy_dokumentow FOR EACH ROW
BEGIN
    :new.ID_Dokumentu := DokumentSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Typy_dokumentow_Dokument_0 AFTER UPDATE OF
ID_Dokumentu
ON Typy_dokumentow FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Dokumentu in table
Typy_dokumentow as it uses sequence.');
```

END;

/

```

-- Trigger for sequence RealizacjaSeq1 for column ID_Realizacji_Wydarzenia in table
Realizacje_Wydarzen -----
CREATE OR REPLACE TRIGGER ts_Realizacje_Wydarzen_Reali_0 BEFORE INSERT
ON Realizacje_Wydarzen FOR EACH ROW
BEGIN
    :new.ID_Realizacji_Wydarzenia := RealizacjaSeq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Realizacje_Wydarzen_Real_0 AFTER UPDATE OF
ID_Realizacji_Wydarzenia
ON Realizacje_Wydarzen FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column ID_Realizacji_Wydarzenia in
table Realizacje_Wydarzen as it uses sequence.');
```

END;

/

```

-- Create foreign keys (relationships) section -----
-----

ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY (ID_Domu) REFERENCES
Domy_Kultury (ID_Domu)
/

ALTER TABLE Sale ADD CONSTRAINT Posiada FOREIGN KEY (ID_Domu) REFERENCES
Domy_Kultury (ID_Domu)
/

ALTER TABLE Wydarzenia ADD CONSTRAINT Oferuje FOREIGN KEY (ID_Domu) REFERENCES
Domy_Kultury (ID_Domu)
/

ALTER TABLE Domy_Kultury ADD CONSTRAINT Dom_kultury_posiada_adres FOREIGN KEY
(ID_Adresu) REFERENCES Adresy (ID_Adresu)
/

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_posiada_adres FOREIGN KEY
(ID_Adresu) REFERENCES Adresy (ID_Adresu)
/

ALTER TABLE Uczestnicy ADD CONSTRAINT Uczestnik_posiada_adres FOREIGN KEY
(ID_Adresu) REFERENCES Adresy (ID_Adresu)
/

ALTER TABLE Adresy ADD CONSTRAINT Adres_posiada_poczte FOREIGN KEY (ID_Poczty)
REFERENCES Poczty (ID_Poczty)
/

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_posiada_stanowisko FOREIGN KEY
(ID_Stanowiska) REFERENCES Stanowiska (ID_Stanowiska)
/

ALTER TABLE Wynagrodzenia ADD CONSTRAINT Ma_wynagrodzenie FOREIGN KEY
(ID_Pracownika) REFERENCES Pracownicy (ID_Pracownika)
/

ALTER TABLE Posiadanie_zezwolenia ADD CONSTRAINT Posiada_zezwolenie FOREIGN KEY
(ID_Pracownika) REFERENCES Organizatorzy (ID_Pracownika)
/

ALTER TABLE Posiadanie_zezwolenia ADD CONSTRAINT Jest_posiadane FOREIGN KEY
(ID_Zezwolenia) REFERENCES Typy_zezwolen (ID_Zezwolenia)
/

ALTER TABLE Posiadanie_specjalizacji ADD CONSTRAINT Posiada_specjalizacje FOREIGN
KEY (ID_Pracownika) REFERENCES Prowadzacy (ID_Pracownika)
/

ALTER TABLE Posiadanie_specjalizacji ADD CONSTRAINT Jest_posiadana FOREIGN KEY
(ID_Specjalizacji) REFERENCES Specjalizacje (ID_Specjalizacji)
/

ALTER TABLE Prowadzacy ADD CONSTRAINT Jest_posiadany FOREIGN KEY (ID_Dokumentu)
REFERENCES Typy_dokumentow (ID_Dokumentu)

```


/

```
ALTER TABLE Realizacje_Wydarzen ADD CONSTRAINT Wydarzenie_ma_realizacje FOREIGN KEY  
(ID_Wydarzenia) REFERENCES Wydarzenia (ID_Wydarzenia)
```

/

```
ALTER TABLE Zapisy_na_wydarzenie ADD CONSTRAINT Jest_uczeszczana FOREIGN KEY  
(ID_Realizacji_Wydarzenia) REFERENCES Realizacje_Wydarzen  
(ID_Realizacji_Wydarzenia)
```

/

```
ALTER TABLE Realizacje_Wydarzen ADD CONSTRAINT Zajmuje FOREIGN KEY (ID_Sali)  
REFERENCES Sale (ID_Sali)
```

/

```
ALTER TABLE Obslugiwanie_wydarzenia ADD CONSTRAINT Realizacja_jest_obslugiwana  
FOREIGN KEY (ID_Realizacji_Wydarzenia) REFERENCES Realizacje_Wydarzen  
(ID_Realizacji_Wydarzenia)
```

/

5.3.2 Uzupełnienie danych

```
INSERT INTO Poczty (Kod_pocztowy, Miasto) VALUES ('00-123', 'Warszawa');  
INSERT INTO Poczty (Kod_pocztowy, Miasto) VALUES ('00-543', 'Warszawa');  
INSERT INTO Poczty (Kod_pocztowy, Miasto) VALUES ('05-473', 'Mińsk  
Mazowiecki');  
INSERT INTO Poczty (Kod_pocztowy, Miasto) VALUES ('40-153', 'Gdańsk');
```

```
INSERT INTO Adresy (Ulica, Nr_domu, Nr_mieszkania, ID_Poczty)  
VALUES ('Chopina', 20, 2, 3);  
INSERT INTO Adresy (Ulica, Nr_domu, Nr_mieszkania, ID_Poczty) VALUES ('Aleje  
Ujazdowskie', 443, 2, 2);  
INSERT INTO Adresy (Ulica, Nr_domu, Nr_mieszkania, ID_Poczty)  
VALUES ('Chopina', 43, 1, 1);  
INSERT INTO Adresy (Ulica, Nr_domu, Nr_mieszkania, ID_Poczty)  
VALUES ('Feliksa Nowowiejskiego', 25, 1a, 4);  
INSERT INTO Adresy (Ulica, Nr_domu, Nr_mieszkania, ID_Poczty)  
VALUES ('Chłopska', 20, 2, 3);  
INSERT INTO Adresy (Ulica, Nr_domu, ID_Poczty) VALUES ('Feliksa  
Nowowiejskiego', 40, 4);  
INSERT INTO Adresy (Ulica, Nr_domu, Nr_mieszkania, ID_Poczty)  
VALUES ('Arbiturów', 3, '4b', 3);
```

```
INSERT INTO Domy_Kultury (Nazwa, Data_zalozenia, Dyrektor, ID_Adresu)  
VALUES ('Nadbałtycki Dom Kultury', TO_DATE('12/01/2000', 'dd/mm/yyyy'),  
'Adam Michałowicz', 1);
```

```
INSERT INTO Stanowiska (Nazwa_Stalowiska, Opis_stalowiska) VALUES  
( 'Nauczyciel języka angielskiego', 'Nauczyciel języka angielskiego na  
poziomie A2');
```

```
INSERT INTO Stanowiska (Nazwa_Stalowiska) VALUES ('Pracownik kina');
```

```

INSERT INTO Stanowiska (Nazwa_Stanowiska) VALUES ('Nauczyciel Tańca
towarzyskiego');
INSERT INTO Stanowiska (Nazwa_Stanowiska, Opis_stanowiska) VALUES
('Nauczyciel języka hiszpańskiego', 'Nauczyciel języka hiszpańskiego na
poziomie A1');

INSERT INTO Uczestnicy (Imie, Nazwisko, Plec, Data_Urodzenia, Email,
ID_Adresu) VALUES ('Adam', 'Wojnarowski', 'Mężczyzna', TO_DATE('14/02/2000',
'dd/mm/yyyy'), 'adamwojnarowski1@gmail.com', 2);
INSERT INTO Uczestnicy (Imie, Nazwisko, Plec, Data_Urodzenia, Email,
ID_Adresu) VALUES ('Michał', 'Wojnarowski', 'Mężczyzna',
TO_DATE('14/02/2000', 'dd/mm/yyyy'), 'michalwojnarowski53@gmail.com', 2);
INSERT INTO Uczestnicy (Imie, Nazwisko, Plec, Data_Urodzenia,
Numer_telefonu, Email, ID_Adresu) VALUES ('Paweł', 'Gortat', 'Mężczyzna',
TO_DATE('29/06/2004', 'dd/mm/yyyy'), '126784567', 'adjkdjksfgkl@wp.pl',
3);
INSERT INTO Uczestnicy (Imie, Nazwisko, Plec, Data_Urodzenia,
Numer_telefonu, Email, ID_Adresu) VALUES ('Julia', 'Wieniawa', 'Kobieta',
TO_DATE('04/12/2008', 'dd/mm/yyyy'),
'456289654', 'kochamkoty432532@gmail.com', 4);

INSERT INTO PRACOWNICY (Imie, Nazwisko, Data_urodzenia, PESEL, Nr_telefonu,
Email, Plec, Data_zatrudnienia, ID_Domu, ID_Adresu, ID_stanowiska) VALUES
('Stefan', 'Baczyński', TO_DATE('14/07/1998',
'dd/mm/yyyy'), '19980714123', '123456789', 'stefanb123@gmail.com',
'Mężczyzna', TO_DATE('14/07/2012', 'dd/mm/yyyy'), 1, 5, 1);
INSERT INTO PRACOWNICY (Imie, Nazwisko, Data_urodzenia, PESEL, Nr_telefonu,
Email, Plec, Data_zatrudnienia, ID_Domu, ID_Adresu, ID_stanowiska) VALUES
('Emila', 'Adamska', TO_DATE('23/08/1997', 'dd/mm/yyyy'), '19970823456',
'789456123', 'anitaolga@gmail.com', 'Kobieta', TO_DATE('06/07/2013',
'dd/mm/yyyy'), 1, 6, 2);
INSERT INTO PRACOWNICY (Imie, Nazwisko, Data_urodzenia, PESEL, Nr_telefonu,
Email, Plec, Data_zatrudnienia, ID_Domu, ID_Adresu, ID_stanowiska) VALUES
('Micha', 'Dworak', TO_DATE('24/06/1996', 'dd/mm/yyyy'),
'19960624456', '598456123', 'kochamkocty@gmail.com', 'Mężczyzna',
TO_DATE('14/06/2014', 'dd/mm/yyyy'), 1, 4, 3);
INSERT INTO PRACOWNICY (Imie, Nazwisko, Data_urodzenia, Email, Plec,
Data_zatrudnienia, ID_Domu, ID_Adresu, ID_stanowiska) VALUES ('Kamil',
'Baczyński', TO_DATE('14/07/1998', 'dd/mm/yyyy'), 'jkrfshjsfjk@gmail.com',
'Mężczyzna', TO_DATE('18/02/2013', 'dd/mm/yyyy'), 1, 5, 3);
INSERT INTO PRACOWNICY (Imie, Nazwisko, Data_urodzenia, Email, Plec,
Data_zatrudnienia, ID_Domu, ID_Adresu, ID_stanowiska) VALUES ('Olga',
'Tuszyńska', TO_DATE('14/09/1995', 'dd/mm/yyyy'), 'wojkte@gmail.com',
'Kobieta', TO_DATE('13/07/2014', 'dd/mm/yyyy'), 1, 3, 4);

INSERT INTO Wynagrodzenia (Data_wyslania_wynagrodzenia, Kwota_wynagrodzenia,
ID_Pracownika) VALUES (TO_DATE('10/10/2020', 'dd/mm/yyyy'), 3000, '1');
INSERT INTO Wynagrodzenia (Data_wyslania_wynagrodzenia, Kwota_wynagrodzenia,
ID_Pracownika) VALUES (TO_DATE('10/10/2020', 'dd/mm/yyyy'), 3200, '2');
INSERT INTO Wynagrodzenia (Data_wyslania_wynagrodzenia, Kwota_wynagrodzenia,
ID_Pracownika) VALUES (TO_DATE('10/10/2020', 'dd/mm/yyyy'), 2200, '3');
INSERT INTO Wynagrodzenia (Data_wyslania_wynagrodzenia, Kwota_wynagrodzenia,
ID_Pracownika) VALUES (TO_DATE('10/10/2020', 'dd/mm/yyyy'), 5200, '4');
INSERT INTO Wynagrodzenia (Data_wyslania_wynagrodzenia, Kwota_wynagrodzenia,
ID_Pracownika) VALUES (TO_DATE('10/10/2020', 'dd/mm/yyyy'), 2600, '5');

```

```

INSERT INTO Sale(Numer_Sali, Maksymalna_Pojemnosc, Czy_Rzutnik,
Czy_Internet, ID_Domu ) VALUES('1', 30, 'T', 'T', 1);
INSERT INTO Sale (Numer_Sali, Maksymalna_Pojemnosc, Czy_Rzutnik,
Czy_Internet, ID_Domu ) VALUES('2', 25, 'T', 'T', 1);
INSERT INTO Sale (Numer_Sali, Maksymalna_Pojemnosc, Czy_Rzutnik,
Czy_Internet, ID_Domu ) VALUES('3', 40, 'T', 'N', 1);
INSERT INTO Sale (Numer_Sali, Nazwa_sali, Maksymalna_Pojemnosc,
Czy_Rzutnik, Czy_Internet, ID_Domu ) VALUES('10','Informatyczna', 40, 'T',
'T', 1);

```

```

INSERT INTO Wydarzenia (Nazwa_wydarzenia, Maksymalna_ilosc_uczestnikow,
Wymagana_zgoda_rodzica, ID_domu) VALUES ('Język Angielski', 25, 'N', 1);
INSERT INTO Wydarzenia (Nazwa_wydarzenia, Maksymalna_ilosc_uczestnikow,
Wymagana_zgoda_rodzica, ID_domu) VALUES ('Nauka Tańca Towarzyskiego', 20,
'N', 1);
INSERT INTO Wydarzenia (Nazwa_wydarzenia, Maksymalna_ilosc_uczestnikow,
Wymagana_zgoda_rodzica, ID_domu) VALUES ('Seans Młodzi Gniewni', 40, 'T',
1);
INSERT INTO Wydarzenia (Nazwa_wydarzenia, Maksymalna_ilosc_uczestnikow,
Wymagana_zgoda_rodzica, ID_domu) VALUES ('Język Hiszpański', 25, 'N', 1);

```

```

INSERT INTO Realizacje_Wydarzen (Data_wydarzenia, Cena, ID_Wydarzenia,
ID_sali) VALUES (TO_DATE('11/12/2020 20:00:00', 'dd/mm/yyyy hh24:mi:ss'),
10.50, 1, 1);
INSERT INTO Realizacje_Wydarzen (Data_wydarzenia, Cena, ID_Wydarzenia,
ID_sali) VALUES (TO_DATE('14/11/2020 18:00:00', 'dd/mm/yyyy hh24:mi:ss'),
10, 2, 2);
INSERT INTO Realizacje_Wydarzen (Data_wydarzenia, Cena, ID_Wydarzenia,
ID_sali) VALUES (TO_DATE('15/12/2020 14:30:00', 'dd/mm/yyyy hh24:mi:ss'),
12.50, 3, 3);

```

```

INSERT INTO Obslugiwanie_Wydarzenia (ID_Pracownika,
ID_Realizacji_Wydarzenia) VALUES (1,1);
INSERT INTO Obslugiwanie_Wydarzenia (ID_Pracownika,
ID_Realizacji_Wydarzenia) VALUES (3,2);
INSERT INTO Obslugiwanie_Wydarzenia (ID_Pracownika,
ID_Realizacji_Wydarzenia) VALUES (2,3);

```

```

INSERT INTO Zapisy_na_wydarzenie (ID_Uczestnika, ID_realizacji_wydarzenia,
Data_zapisu) VALUES (1, 1, TO_DATE('08/10/2020', 'dd/mm/yyyy'));
INSERT INTO Zapisy_na_wydarzenie (ID_Uczestnika, ID_realizacji_wydarzenia,
Data_zapisu) VALUES (2, 1, TO_DATE('08/10/2020', 'dd/mm/yyyy'));
INSERT INTO Zapisy_na_wydarzenie (ID_Uczestnika, ID_realizacji_wydarzenia,
Data_zapisu) VALUES (3, 2, TO_DATE('08/10/2020', 'dd/mm/yyyy'));
INSERT INTO Zapisy_na_wydarzenie (ID_Uczestnika, ID_realizacji_wydarzenia,
Data_zapisu) VALUES (4, 3, TO_DATE('08/10/2020', 'dd/mm/yyyy'));

```

```

INSERT INTO Typy_dokumentow (Nazwa_Dokumentu) VALUES ('Dyplom ukonczenia
szkoy językowej');
INSERT INTO Typy_dokumentow (Nazwa_Dokumentu) VALUES ('Dyplom ukonczenia
szkoy tańca');
INSERT INTO Typy_dokumentow (Nazwa_Dokumentu) VALUES ('Dyplom ukonczenia
szkoy filmowej');

```

```

INSERT INTO Specjalizacje (Nazwa_Specjalizacji) VALUES ('język agnielski');
INSERT INTO Specjalizacje (Nazwa_Specjalizacji) VALUES ('język
hiszpanski');

```

```

INSERT INTO Specjalizacje (Nazwa_Specjalizacji) VALUES ('Nauczyciel
tanga');
INSERT INTO Specjalizacje (Nazwa_Specjalizacji) VALUES ('język rosyjski');

INSERT INTO Typy_Zezwolen (Nazwa_Zezwolenia) VALUES ('Pozwolenia na
organizacje wydarzeń masowych');
INSERT INTO Typy_Zezwolen (Nazwa_Zezwolenia) VALUES ('Pozwolenia do ochrony
budyńku');

INSERT INTO Prowadzacy (ID_Pracownika, Wyksztalcenie, Nr_dokumentu,
Data_wydania_dokumentu, ID_Dokumentu) VALUES (1, 'wyzsze', '45678946a',
TO_DATE('11/01/2020', 'dd/mm/yyyy'), 1);
INSERT INTO Prowadzacy (ID_Pracownika, Wyksztalcenie, Nr_dokumentu,
Data_wydania_dokumentu, ID_Dokumentu) VALUES (5, 'wyzsze', '45654f6a',
TO_DATE('12/01/2020', 'dd/mm/yyyy'), 1);
INSERT INTO Prowadzacy (ID_Pracownika, Wyksztalcenie, Nr_dokumentu,
Data_wydania_dokumentu, ID_Dokumentu) VALUES (3, 'wyzsze', '4dfgdsfa',
TO_DATE('01/01/2020', 'dd/mm/yyyy'), 2);
INSERT INTO Prowadzacy (ID_Pracownika, Wyksztalcenie, Nr_dokumentu,
Data_wydania_dokumentu, ID_Dokumentu) VALUES (4, 'srednie', '4dfgffff6a',
TO_DATE('02/12/2010', 'dd/mm/yyyy'), 2);

INSERT INTO Posiadanie_Specjalizacji (ID_Pracownika, ID_Specjalizacji)
VALUES (1, 1);
INSERT INTO Posiadanie_Specjalizacji (ID_Pracownika, ID_Specjalizacji)
VALUES (5, 2);
INSERT INTO Posiadanie_Specjalizacji (ID_Pracownika, ID_Specjalizacji)
VALUES (3, 3);
INSERT INTO Posiadanie_Specjalizacji (ID_Pracownika, ID_Specjalizacji)
VALUES (4, 3);

INSERT INTO Organizatorzy (ID_Pracownika, Nr_zezwolenia,
Data_wydania_zezwolenia) VALUES (2, '5465456pa', TO_DATE('03/12/2019',
'dd/mm/yyyy'));

INSERT INTO Posiadanie_Zezwolenia (ID_Pracownika, ID_Zezwolenia) VALUES
(2, 1);

```

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

- Ile osób zapisało się na daną realizację wydarzenia:

```
SELECT COUNT(*) from Zapisy_na_wydarzenie
WHERE id_realizacji_wydarzenia=1;
```

- Pracownicy mieszkający w Warszawie:

```
SELECT ID_pracownika, Imie, Nazwisko from Pracownicy
JOIN Adresy ON Pracownicy.ID_Adresu = Adresy.ID_Adresu
JOIN Poczty ON Adresy.ID_Poczty = Poczty.ID_Poczty
WHERE Poczty.Miasto = 'Warszawa';
```

- Informacje o pracownikach (imie, nazwisko, stanowisko oraz email), którzy obsługują daną realizację wydarzenia

```
SELECT pracownicy.imie, pracownicy.nazwisko,
stanowiska.nazwa_stanowiska, pracownicy.email from
Obslugiwanie_wydarzenia
JOIN Pracownicy ON
Pracownicy.ID_Pracownika=Obslugiwanie_wydarzenia.ID_Pracownika
JOIN Stanowiska
Pracownicy.ID_Stnowiska=Stanowiska.ID_stnowiska
WHERE Obslugiwanie_wydarzenia.id_realizacji_wydarzenia=1;
```

- Wydarzenia jakie oferuje dom kultury, gdzie potrzebna jest zgoda rodzica:

```
SELECT id_wydarzenia, nazwa_wydarzenia from wydarzenia
WHERE Wymagana_zgoda_rodzica='T';
```

- Daty realizacji wydarzeń konkretnego wydarzenia:

```
SELECT Data_wydarzenia, Wydarzenia.nazwa_wydarzenia from
Realizacje_wydarzen
JOIN Wydarzenia ON
Realizacje_wydarzen.Id_wydarzenia=Wydarzenia.Id_wydarzenia
WHERE Wydarzenia.Id_wydarzenia=2;
```

6. Bibliografia

- Wykłady oraz materiały przygotowane przez prowadzącego dr. hab. Marcin Kowalczyk
- <https://docs.oracle.com/database/121/SQLRF/toc.htm>
- <https://www.w3schools.com/sql/>
- <http://hilite.me/>

6.1 Model logiczny po procesie denormalizacji

